Haskell Code Sample Reference: (source: <https://pinetools.com/syntax-highlighter>)

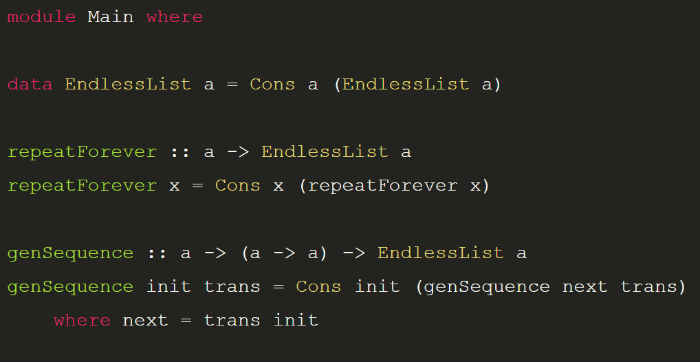
**Text

Description automatically generatedText

Description automatically generatedZenburn** **xt 256**

Text

Description automatically generated**Qtcreator Dark** **Monokai Sublime**



**Original Color Scheme (C++):**

#include <iostream>

#include <math.h>

using namespace std;

int main(){

double x = 2.3; //Comments

cout << “Hello there “ << sin(x) << endl;

return 0;

|  |  |
| --- | --- |
| **Color** | **Haskell Category** |
| #F6F6F6 (White) | Regular Text |
| #1E1E1F (Black) | Background |
| #CACCD2 (Light Gray) | Comments |
| #EA428F (Rose) | Params |
| #E9402F (Red) | Numbers |
| #DFFD52 (Yellow) | Function Name |
| #AEFA4E (Lizard Green) | Chars/Strings |
| #74FBAF (Spring Green) | def, class def |
| #66DDEF (Blue) | import, while, assignments (=)?, ptrs (\*)? |

Haskell Code Sample Reference

import Control.Monad

import Text.ParserCombinators.Parsec

-- AST Definition

data CALC where

  Num :: Int -> CALC

  Plus :: CALC -> CALC -> CALC

  Minus :: CALC -> CALC -> CALC

  deriving (Show,Eq)

-- Exercise 1

evalErr :: CALC -> Int

evalErr (Num x) = if x<0

      then error "Bad no don't!"

                  else x

evalErr (Plus l r) = ((evalErr l) + (evalErr r))

evalErr (Minus l r) = let x = ((evalErr l) - (evalErr r)) in

                    if x<0

                    then error "Bad no don't!"

                    else x

parseString p str =

    case parse p "" str of

    Left e -> error $ show e

    Right r -> r

main = do

  putStrLn "Hello"

  putStrLn "World"